Date: 13/03/2014

Parag R Adhau 120101049 Gaurav Madhuker 120101028

ROM

Internal memories are semiconductor memory. Semiconductor memories are categories as volatile memory and non-volatile memory.

RAM: Random Access Memories are **volatile** in nature. As soon as the computer is switched off, the contents of memory are also lost.

ROM: Read only memories are **nonvolatile** in nature. The storage is permanent, but it is read only memory. We cannot store new information in ROM.

## Different types of rom

Several types of ROM are available:

* PROM: Programmable Read Only Memory; it can be programmed once as per user requirements.
* EPROM: Erasable Programmable Read Only Memory; the contents of the memory can be erased and store new data into the memory. In this case, we have to erase whole information.
* EEPROM: Electrically Erasable Programmable Read Only Memory; in this type of memory the contents of a particular location can be changed without effecting the contents of other location.

**Basic architecture of ROM using decoder:** A read – only memory is essentially a memory device in which permanent binary information is stored. The binary information must be specified by the designer and is then embedded in the unit to form the required interconnection pattern. Once the pattern is established, it stays within the unit even when power is turned off and on again.

Consider the given below example a 16\*8 ROM. The unit consists of 16 words of 8 bits each. There are four input lines that from binary numbers from 0 through 15 for the address. And this figure shows the internal logic construction of the ROM. A programmable connection between two lines is logically equivalent to a switch that can be altered to either be close (meaning that two lines are connected) or open (meaning that the two lines are disconnected). The programmable intersection between two lines is called a cross point. Various physical devices are used to implement cross point switches. One of the simplest technologies employs a fuse that normally connects the two points, but is opened or "blown” by applying a high voltage pulse into the fuse. In this architecture OR gate side is programmable & AND gate side is fix. This architecture is not best because we have to do all computation for each input.

The internal binary storage of a ROM is specified by a truth table that shows the word content in each address. For example, the content of 16\*8 ROM may be specified by a truth table similar to the one which is shown in given below table . The truth table shows the four inputs under which are listed 16 addresses. At each address, there is stored a word of 8 bits, Which is listed under the output columns. The table shows only the first four and last three four words in the ROM.

The hardware procedure that programs the ROM results in blowing fuse links according to given truth table. In below given example, every 0 listed in the truth table specifies a no connection and every 1 listed specifies a path that is obtained by a connection. For example, the table specifies the 8-bit word 00100100 for permanent storage at address 1. The six 0’s in the word are programmed by blowing the fuse links between the output 1 of the decoder and the inputs of OR gate associated with outputs A7, A6, ………..A1 and A0. The two 1’s in the words are marked in the diagrams with a (.) to denote a connection .The signal propagates through the connections to the OR gate outputs of A5 and A3. The other six remains at 0. The results is that the stored word 00100100 is applied to 8 data outputs.

Input Output



BASIC ARCHITECTURE OF ROM

**PLA (programmable logic array):** PLAs are nothing but the arrays of AND gates followed by array of OR gates.  If the control signals are expressed as sum of product form then with the help of PLA it can be implemented. In PLA truth table is minimized with karnaugh map using sum of product form. In PLA set of AND gate linking to OR gate.

The internal logic of a PLA with three inputs and two outputs is shown below. Each input goes through a buffer and an inverter shown in the given diagram. Each inputs and its complement are connected to the inputs of each and gate as indicated by the intersection between the vertical and horizontal lines. The outputs of each AND gate is connected to inputs of each OR gate goes to XOR gate where the other input can be programmed to receive a signal equal to either 1 or 0. The output is inverted when XOR input is connected to 1. The output does not change when the XOR input is connected to 0. The particular Boolean functions implemented in the given example are

F1 = AC + AB’C’

F2 = ( A’B’ + ABC’)’

The product terms generated in each AND gates are listed along the output of each gate in the diagram. The product term is determined from the inputs whose cross points are connected with a (.) dot. The output may be complemented or left in its true form depending on the connection of the XOR gate inputs.



PROGRAMMABLE LOGIC ARRAY

**PROGRAMMABLE ARRAY LOGIC (PAL):** PAL is programmable logic device with a fixed OR array and programmable AND array. Because only the AND gates are programmable. The PAL is easier to program, but is not as flexible as the PLA. It has four inputs and four outputs. Each input has a buffer –inverter gate and each output is generated by a fixed OR gate. Each AND gate has 10 programmable input connections. This is shown in diagram by 10 vertical lines intersecting each horizontal line. The horizontal line symbolizes the multiple-input configurations of the AND gate, One of the outputs is connected to a buffer inverter gate and then feedback into two inputs of the AND gates. Boolean functions are

W = ABD’ + A’B’C’D

Z = W + AC’D’ + A’B’CD’

**Difference between PAL, PLA and basic architecture of ROM :**

Basic architecture of ROM : In this architecture OR gate side is programmable & AND gate side is fix. This architecture is not best because we have to do all computation for each input.

PLA : In this architecture both OR gate and AND gate side is programmable . It is better in compare to basic architecture because in this architecture we don’t need do all computation because we can minimize our input by karnaugh map.

PAL: In this architecture AND gate side is programmable & OR gate side is fix. It is easier to program, but PLA is not flexible.



PROGRAMMABLE ARRAY LOGIC